

# Manipulation, Learning, and Recall with Tangible Pen-Like Input

Lisa A. Elkin<sup>2,1</sup>, Jean-Baptiste Beau<sup>1</sup>, G ry Casiez<sup>3,4,5,1</sup>, Daniel Vogel<sup>1</sup>

<sup>1</sup>School of Computer Science, University of Waterloo, Waterloo, Canada; <sup>2</sup>Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, United States; <sup>3</sup>Univ. Lille, UMR 9189 - CRISTAL, Lille, France; <sup>4</sup>Inria, France; <sup>5</sup>Institut Universitaire de France (IUF)  
{jalbeau, dvogel}@uwaterloo.ca, lelkin@cs.washington.edu, gery.casiez@univ-lille.fr

## ABSTRACT

We examine two key human performance characteristics of a pen-like tangible input device that executes a different command depending on which corner, edge, or side contacts a surface. The manipulation time when transitioning between contacts is examined using physical mock-ups of three representative device sizes and a baseline pen mock-up. Results show the largest device is fastest overall and minimal differences with a pen for equivalent transitions. Using a hardware prototype able to sense all 26 different contacts, a second experiment evaluates learning and recall. Results show almost all 26 contacts can be learned in a two-hour session with an average of 94% recall after 24 hours. The results provide empirical evidence for the practicality, design, and utility for this type of tangible pen-like input.

## Author Keywords

pen input; tangible interfaces; learning; command selection.

## CCS Concepts

•Human-centered computing → Graphical user interfaces; Graphics input devices; Empirical studies in HCI;

## INTRODUCTION

There is a class of tangible input devices that not only provide positional input, but also detect which geometric feature contacts a surface. This class has an advantage over traditional input methods because it integrates command selection (how it contacts) and parameter manipulation (positional movement). A common example is a digital stylus which uses a nib end and an eraser end to switch between drawing and erasing. Extending this approach has been demonstrated in research like ToolStone [18], and a pen-like input device called Cont  [27].

Cont  is held in one hand, and manipulated so that one of the 8 corners, 12 edges, or 6 sides of its cuboid shape contacts a tablet (Figure 1). In theory, these 26 different contacts enable selecting among 26 different commands. However, the

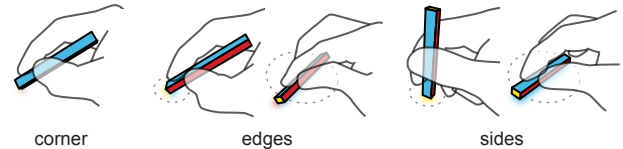


Figure 1. Cont , a tangible pen-like input device (from [27]).

prototype only sensed 10 different contacts and required a diffuse illumination table top. No studies were conducted, so implicit claims by the authors that devices like Cont  would be fast to manipulate and users could remember many contact-to-command mappings have never been evaluated.

Testing these manipulation and recall claims using a 26-command Cont  device contributes fundamental knowledge about tangible pen-like devices, general tangible input [8, 2, 18], and 3D spatial memory for command selection [6, 19]. In this paper we examine these questions using the simple cuboid shape used for Cont , a logical and essential first step before examining other variations of 3D shapes and deeper investigations of underlying phenomena.

We test manipulation and recall in two controlled experiments, and we describe a new hardware device to sense all 26 contacts on a standard touchscreen. Our first experiment examines manipulation by timing people as they rotate the device with their fingers or hand to move from one contact to another. Three device sizes are examined, with a pen mockup included as a baseline. Although the original work suggested small form factors would be faster to manipulate, we find the largest size is fastest overall. We also find most manipulation times are equivalent to transitions with the pen.

We designed and created a new self-contained Cont  prototype that can sense all 26 contacts and works with any capacitive display using a conductive case designed with pliable corners. Contacts are distinguished using the device angle from an internal IMU, and a 3D “mirror” visualization displays a re-configurable mapping of commands to contacts. This device enabled us to study recall in a second experiment based on Gutwin et al.’s [5] protocol. An adaptation of FastTap [6] is used as a baseline comparison where commands are spatially anchored in 2D, compared to 3D with the Cont  device. Our results show that after approximately 2 hours of training, people can recall 24 out of 26 Cont  commands 24 hours later.

Our work makes three contributions: (1) manipulating a tangible input device like Conté can be fast, and speed increases with device size; (2) people can recall many command mappings when spatially mapped to a 3D object, and this number is comparable to a 2D spatial mapping; (3) a Conté device that senses and distinguishes all 26 contacts can be fabricated for current capacitive displays.

## BACKGROUND AND RELATED WORK

A tangible input device is an object coupled with digital information [8]. They can be “graspable” to take advantage of spatial reasoning, existing manipulation skills, and a one-to-one mapping between control and controller [2]. Fitzmaurice et al.’s *Bricks* is a classic example of a tangible input device. These Lego-sized extruded-rectangles act as graspable physical handles for direct manipulation of virtual objects. Rekimoto and Sciammarella [18] extend this idea to selecting six menus or modes based on which side of their mouse-sized *Toolstone* contacts the surface. Manipulation time was not investigated, and recall was not considered since the static mapping is labelled on the large sides of the device. Van Laerhoven et al. [26] describe methods to sense orientation and movement of a cube-shaped device. One example of contact-based input is described, where an audio-input mode is determined based on which side the cube rests. Manipulation time was not investigated, and like ToolStone, recall was not tested since the large device could be labelled with the static mapping. Perelman et al. demonstrate a phone case with unique patterns to detect which side contacts a tablet, but neither edges or corners are considered [15]. Our prototype extends these general approaches of using orientation to detect how a device contacts a surface.

### *Vogel and Casiez’s Conté* [27]

Built to resemble the artists’ square-sided crayon of the same name, the thin extruded-rectangle shape affords a pen-like feeling when a corner or end edge contacts a surface. Other edges and sides may be used for interactions like simulating a mouse, tool pallets, or context menus. When describing manipulation characteristics, the authors argue “Many nearby transitions should be quite fast.” but they admit “In our [two-person] exploratory interview, participants commented on the extra time required to flip Conté end-over-end” [27, p. 360]. In addition to these untested and unproven statements about manipulation, the authors state “An open question is whether a user could recall command mappings for 26 different contact points” [27, p. 360]. Our work investigates exactly these issues related to manipulation time and recall.

A possible reason why such fundamental questions were not examined may be the limited capability of their hardware prototype. The device emitted infrared (IR) light from each corner and was covered in IR reflective paper with dot patterns on each side. When combined with a diffuse illumination tabletop, computer vision techniques could determine the device’s position, and distinguish between differently-shaped contacts. For example, the short, medium, and long edge could be distinguished, but edges of the same size could not be distinguished. Similarly, all eight corners appeared the same. As a result,

only 10 out of 26 possible contacts could be identified. Our device demonstrates that distinguishing between all 26 contacts is possible without external tracking hardware or cameras.

## Device Manipulation Characteristics

A simple form of contact-based input is flipping between “ink” and “eraser” ends of a digital pen [7, 13]. Li et al. investigated this manipulation time in a larger study on general pen mode-switching [12]. They found flipping the pen slower than alternatives like pushing a button, but they did not control for device size, considered techniques with few mode-switches, and only this simple end-to-end transition was tested. Other projects related to pen manipulation include using pen grip changes for mode-switching, like Song et al. [21] and Sun et al. [22]. However, manipulation times are not evaluated, and these interactions are not contact-based.

Many factors influence how a device is gripped and manipulated. For instance, Olafsdottir et al. found grip is influenced by hand position, target position, and anticipated rotation when manipulating graphical objects on a tabletop [14]. However these are not tangible objects, and they do not control for object size. When designing the *Roly-Poly Mouse* [16], Perelman et al. refer to informal tests that identified an 8 cm size device was easier to manipulate compared to 6 cm or 10 cm. A detailed examination of manipulation time for tangible input devices has not been performed.

## Learning and Recall for Spatially Mapped Interfaces

Designing an interface to be spatially consistent leverages spatial memory so item locations are retrieved with little effort [19]. This design principle has been adopted in many interfaces; we focus on *FastTap* [6] since it has been extensively tested and we use an adapted version as a baseline.

*FastTap* is a 2D grid of spatially anchored command buttons. A command is invoked by pressing a menu button in one corner, then pressing a grid location. It is a rehearsal interface [10], where the grid is displayed after a short delay (150 to 250 ms [5, 11]). A novice user waits to see the grid of command locations, but once learned, they use “expert mode” by pressing a grid location immediately. In a time-constrained training game, participants quickly transitioned to expert selections with a 12-item grid [5, 11], and expert usage persisted with a realistic application over the course of one week [11]. However, expert use was infrequent with a 24-item grid [5], but this was only tested in a real application. It remains unclear if this is due to the increase in number of items, or the lack of a preparatory training game.

A contact-based tangible device like Conté is essentially a 3D spatial interface, since commands are mapped to landmarks defined by its shape. Perrault et al. [17] tested a 3D spatial interface with commands mapped to objects in a room, often with strong semantic relationships, and found 47 out of 48 commands could be recalled after one week. This is encouraging, but unlike being surrounded by commands in 3D, a Conté device is itself manipulated in 3D, so its spatial frame-of-reference constantly rotates and translates. In 2D, the benefits of spatial consistency are robust to many transformations, like view-aligned translation, but large rotations and perspective

changes incur significant “reorientation” times [20]. Whether 3D spatial mapping is similarly affected is unknown.

Understanding upper limits for learning spatial command locations is important, since some interfaces support up to 64 [24] or 160 items [25]. Aside from Perrault et al., only a small subset are usually tested, typically 12 (e.g. [24]), and formal methods to assess learning, such as memory recall after 24 hours [1], have not been used.

## 26-CONTACT CONTE DEVICE

This section describes our 26-contact capacitive Conté device (Figure 2). It retains the cuboid shape of Vogel and Casiez’s simple 10-command prototype, but its design and implementation are different in all other aspects. Our device is  $85 \times 30 \times 15$  mm to accommodate a microcontroller, IMU, and battery, which is larger than the previous device, and weighs approximately 25g (comparable to a 21g Apple Pencil). We show this increase in size has advantages in our manipulation experiment. Unlike the previous device, ours senses 26 contacts on a conventional capacitive touch surface, the first device of its kind to do so. It is not a polished consumer product, but with routine recalibration, it is good enough to demonstrate the potential of the hardware approach, and allow users to learn command locations in our recall experiment. 3D drawings of parts, schematics, and software are available for replication<sup>1</sup>.

### Case Design

The main case is printed in PLA wrapped in copper tape, and the ends are printed with *Conductive 95 Shore A Thermoplastic Polyurethane (TPU)*<sup>2</sup>. The end corners are designed with hollow pockets to make them pliable enough to register on standard capacitive sensors, but stiff enough to feel like corners. The narrow sides of each end project 1.5mm forming 11mm wide valleys. When combined with the narrower main case, this creates patterns of one, two, or four capacitive touches when corners, edges, or sides are contacting the display. White marks are added to one end to create visual asymmetry, so different orientations are distinguishable.

### Internal Hardware

Orientation, acceleration, and magnetic field data from the 9 DOF MPU9250 IMU are processed by an on-board sensor-fusion chip<sup>3</sup> at 100 Hz, then passed to an Arduino-compatible ESP8285 microcontroller which packages the data, then sends it as a UDP stream over on-board WIFI at 100 Hz. All internal hardware is powered by a 3.7V, 150mAh LiPo battery.

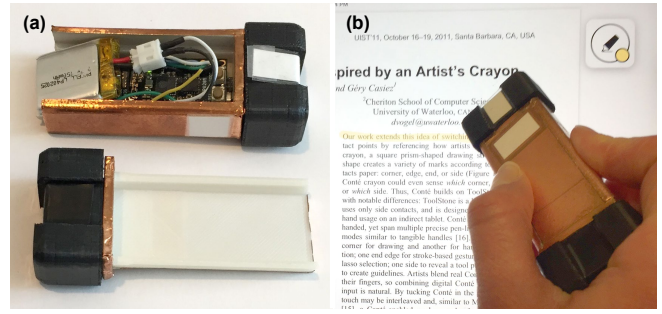
### Software

A host application, written in Java with Processing, combines the UDP stream of IMU data with a TUIO stream of touch events from a standard touch input device to create an OSC stream of device events. The events include *down*, *move*, and *up* events to describe which corner, edge, or side is contacting the display, the  $(x, y)$  position(s) of the touch points, and raw IMU data for custom usage. In the case of two or four touch

<sup>1</sup>ns.inria.fr/loki/conte

<sup>2</sup>rubber3dprinting.com/pi-etpu-95-250-carbon-black/

<sup>3</sup>indie.com/products/onehorse/ultimate-sensor-fusion-solution-mpu9250



**Figure 2. Conductive 26-contact sensing Conté device: (a) internal hardware and case; (b) used in an iOS iPad PDF application.**

points (most edges and sides), a single down and up event is sent when the first or last touch point is registered. In addition, an air event sends IMU data when the device is not contacting the touchscreen.

Generating these events requires touch filtering and contact classification.

When the device is dragged quickly, the reported touchscreen events can “skip” with spurious up and down events, leaving gaps in what should be a continuous line. To counteract this, up and down touchscreen events are ignored when they occur within 300ms and within a distance threshold (46mm for single points, 15mm for two or more points). An exception is made when more than 75% of the device acceleration is along the normal vector of the display, then the anti-skip filter is ignored and the touchscreen up event is immediately accepted. All values were determined by trial and error, to minimize spurious events with reasonable latency.

The contact is classified from IMU data using a decision tree. Training data was gathered by touching the device to a touchscreen 40 times per contact, each time at a different orientation over the possible range. Three frames of IMU readings were recorded after each touch down. This was performed by the first author, since this is hardware calibration to determine the relative orientation of the IMU to the touchscreen. The resulting 3,120 data points were used to train a decision tree in Weka with 10-fold cross-validation, resulting in 99.2% accuracy. During actual usage, the host application uses the trained decision tree to classify the three IMU data frames after a down event, using a majority voting scheme with the last frame acting as tie breaker. Note IMU “drift” and magnetic interference reduce this ideal accuracy during real usage. During Experiment 2, participants performed a drift-reset after about 4.7 minutes of intense usage. Even with this precaution, there are still some misclassifications, most often with adjacent contacts, like a corner and short end edge. We manually correct these errors using video logs for experiment recall tests. Professional-grade engineering and IMU hardware would reduce the need for manual drift correction substantially.

Using a dedicated host to generate events means applications can run on any platform. We demonstrate Conté with an iPad in the accompanying video, and our recall experiment runs on a Windows touchscreen laptop.

## EXPERIMENT 1: MANIPULATION

The goal of this experiment is to investigate the impact of device size on manipulation time: the fundamental tangible interaction of rotating the device with the fingers or hand to change how the device contacts a surface. Understanding manipulation time validates our larger hardware device and guides designers when mapping commands to contact types and locations. Device mock-ups with colour-based detection are used to avoid unnecessary engineering of custom hardware to replicate our device in very small form factors. Overall, we expected larger devices would take longer to manipulate, and that manipulation time would increase as the location of a contact was “farther away” — meaning rotating to a contact on the same end would be fastest, rotating to a contact along the barrel would be slower, and rotating to a contact at the opposite end would be slowest. Since Conté is considered “pen-like”, a pen mock-up is used as a baseline to compare with equivalent Conté contacts.

### Participants

We recruited 12 participants, mean age 29 ( $SD = 7.6$ ), 6 men, 6 women, all right-handed. Left handed people were screened out to avoid asymmetric confounds. Each participant received \$15 for the 90 minute study.

### Apparatus

The Conté device mock-ups were 3D printed in PLA and each face painted with a unique colour in a consistent pattern (Figure 3a). The pen mock-up was a “paper blending stump” with unique colours on the ends and around the barrel. The experiment was performed on a light box with a partially covered acrylic top with a transparent resistive touch sensor placed above an internal video camera (Figure 3b). When a mock-up touches the  $10 \times 4$  cm window in the light box, the camera captures a frame and a hue-based classifier determines which contact was used based on the relative proportion of colours visible from below. The ideal diffuse illumination created by the light box made contact classification very reliable and the resistive display provided accurate timing.

### Procedure

Before explaining the task procedure, we introduce terminology and concise notation. A *manipulation* is when the device is rotated from touching one contact against the screen, to touching another contact against the screen, and back again. With

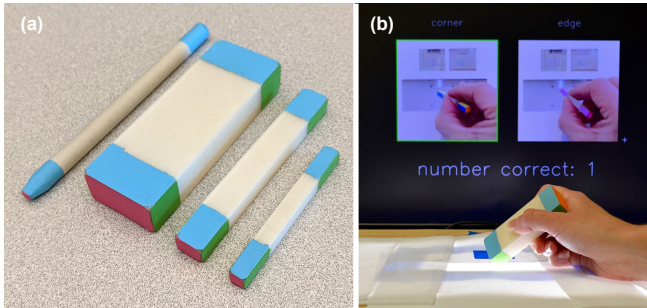


Figure 3. Apparatus: (a) device mock-ups for PEN, LARGE, MEDIUM, SMALL; (b) light table used to detect contact.

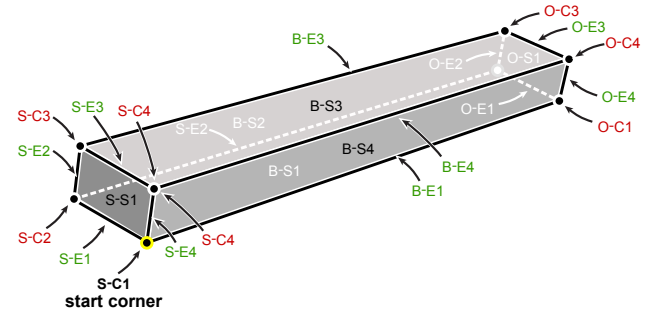


Figure 4. Contact names and groups. Prefix denotes GROUP: ‘S’ means the contact is at the *same end* as the start corner, ‘B’ means the contact is located along the *barrel*, and ‘O’ means the contact is located at the *other end* relative to the start corner. Suffix denotes the contact type and number: ‘C’ for corner, ‘E’ for edge, ‘S’ for side.

26 contacts, there are 325 (26 choose 2) such manipulations. We reduce that number by focusing only on manipulations starting from a specific corner. This is reasonable since Conté is primarily a pen-like device, where corners are used for modes like writing or drawing, and edges and sides used for interspersed commands. In the experiment, each manipulation goes from a “start corner” to one of the 7 remaining corners, or to one of the 12 edges, or to one of the 6 sides, or to the user’s finger, and back to the start corner.

We use a concise notation to describe a *contact* relative to the start corner (illustrated in Figure 4). The prefix denotes the relative position: ‘s’ means the contact is at the *same end*, ‘b’ means the contact is along the *barrel*, and ‘o’ means the contact is at the *opposite end*. The suffix denotes the type of contact (‘c’ for corner, ‘e’ for edge, ‘s’ for side) with a numeric id. For example, ‘s-c2’ is corner 2 at the same end, ‘b-s1’ is side 1 of the barrel, and ‘o-e3’ is edge 3 at the opposite end. The prefixes ‘s’, ‘b’, and ‘o’ form contact location groups.

### Task

The task required participants to perform manipulations in rapid succession, and they had to repeat two error-free manipulations in a row to complete the task. The start corner and other contact in the manipulation were shown in side-by-side images on the screen located near the light table. The image of the active contact was highlighted in green and sound communicated successful and unsuccessful manipulations based on the recognized contact. Each participant was given brief instructions on how to interpret on-screen images and how to complete the task. Participants were instructed to hold the mockups like a pen when possible and told to complete each manipulation as fast as possible. For each manipulation contact, they could practice before actual trials.

### Design

The primary independent variable is device SIZE with three levels: SMALL ( $63 \times 6 \times 6$  mm), the size of an artist’s Conté crayon; MEDIUM ( $84 \times 11 \times 8$  mm), the size of the Vogel and Casiez’s Conté device; and LARGE ( $85 \times 30 \times 15$  mm), the size of our Conté hardware device. The second independent variable is CONTACT, the contact the manipulation uses other



than the start corner. It is an unbalanced factor due to the topology of the mock-ups tested. For MEDIUM and LARGE, there are 25 unique contact manipulations, but due to symmetry in the square-ended SMALL mock-up, only the 16 unique contact manipulations were tested. A manipulation between the start corner and touch is also tested for each mock-up (a related interaction highlighted by Vogel and Casiez), thus touch is included as a CONTACT.

In addition, a mock-up of a digital pen (7 mm diameter, 110 mm length) was tested. It has 5 contacts (nib end, eraser end, cylindrical barrel, and 2 edges separating the ends from the barrel) creating 4 CONTACT manipulations starting from the edge between the nib and barrel. Touch input was also included as a fifth CONTACT.

The experiment design is within-subjects. The combined presentation order for SIZE and the pen condition was counter-balanced using a Latin square, with CONTACT in random order. The dependent variable is *Manipulation Time*, which represents the total time to complete one manipulation.

## Results

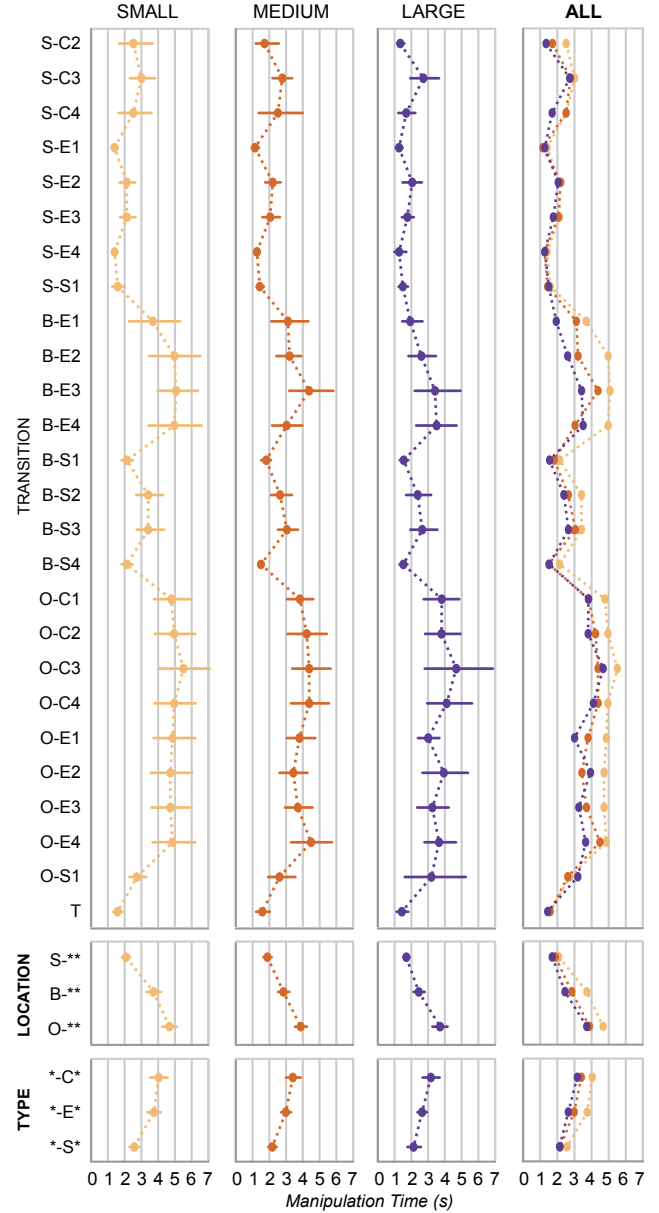
To simplify analysis, data from the 9 symmetric CONTACTS in the SMALL condition are also used for their counterparts, so all SIZES have the same number of data points. We examined manipulation times for the two consecutive error-free manipulations to identify outliers for each combination of SIZE and CONTACT, but did not remove any of the 1656/1872 data points classified as outliers.

Although we do not focus on errors, for context, a type of error measure can be calculated as the mean number of manipulation attempts before performing the two consecutive correct manipulations required for our primary analysis. These error measures were 3.3 for SMALL, 1.6 for MEDIUM, 1.2 for LARGE, and 1.0 for the digital pen mockup.

In the analysis to follow, data with log-normally distributed residuals is log-transformed. A repeated measures ANOVA is performed for balanced designs, and a linear mixed effects (LMM) ANOVA is used for unbalanced designs [3, 4]. Both are followed by post hoc paired t-tests. An ANOVA based on the Aligned Rank Transform (ART) [28] is used when residuals do not follow a known distribution or violate the homoscedasticity assumption of an LMM, and are followed by post hoc Wilcoxon signed rank tests. Holm's sequential Bonferroni procedure is used to correct for multiple comparisons in all post hoc tests.

### Learning Effect

To determine if a learning effect is present, a two-level factor is created to separate manipulation data points into FIRST and SECOND, representing the order of the two consecutive error-free manipulations. We found a significant effect of this factor ( $F_{1,11} = 27.0$ ,  $p < .001$ ,  $\eta_G^2 = .007$ ) on log-transformed *Manipulation Time*, with SECOND (2.9s) faster than FIRST (3.0s). However, considering the small effect size, we aggregated both manipulation times.



**Figure 5. Manipulation Time by CONTACT, contact LOCATION, and contact TYPE** (see Figure 4 for key to contact notation, T means manipulation with touch input). Dashed connecting lines for readability, all factors are categorical.

### Manipulation Time by Device Size

We expected overall *Manipulation Time* would increase with SIZE, but surprisingly, LARGE was fastest. A repeated measures ANOVA found a main effect of SIZE on log-transformed *Manipulation Time* ( $F_{2,22} = 8.9$ ,  $p < .01$ ,  $\eta_G^2 = .1$ ), with post hoc tests finding SMALL (3.47s, SD = 2.27s) slower than MEDIUM (2.86s, SD = 1.81s) and LARGE (2.62s, SD = 2.00s), and MEDIUM slower than LARGE (all  $p < .0001$ ).

### Manipulation Time by Contact

The overall pattern of time by CONTACT appears similar regardless of SIZE (Figure 5). An ANOVA with ART found a

main effect of CONTACT ( $F_{25,847} = 47.7, p < .0001$ ), but no SIZE  $\times$  CONTACT interaction. Across all three SIZES, six of the eight fastest (S-E1, S-E4, S-S1, B-S4, B-S1, T) and five of the eight slowest CONTACTS (O-E4, O-C2, O-C4, B-E3, O-C3) are the same, but their order in terms of time is not identical between all three SIZES. Post-hoc analysis found 203 significantly different pairs of CONTACTS. Compared to all other CONTACTS, touch ('T') is fourth fastest (1.53s).

#### Manipulation Time by Contact Location

To conduct this analysis, a 3-level LOCATION factor is created by grouping CONTACTS by prefix: *same end* ('S-\*\*\*' contacts); *barrel* ('B-\*\*\*' contacts); and *opposite end* ('O-\*\*\*' contacts). An LMM ANOVA on log-transformed *Manipulation Time*, with LOCATION and SIZE as fixed variables and participant id as a random variable, found a main effect for LOCATION ( $F_{2,880} = 41.8, p < .0001$ ) but no significant LOCATION  $\times$  SIZE interaction. Post-hoc comparisons found significant differences ( $p < .0001$ ) between all levels, confirming our expectation that manipulations with nearby contacts are fastest (*same end*: 1.89s, SD = 1.18s), followed by contacts along the middle of Conté (*barrel*: 3.00s, SD = 1.97s), and slowest when far away (*opposite end*: 4.09s, SD = 2.25s) (see bottom of Figure 5).

#### Manipulation Time by Contact Type

Similar to above, a 3-level TYPE factor is created by grouping CONTACTS by suffix: *corners* ('\*-C\*') contacts); *edges* ('\*-E\*') contacts); and *sides* ('\*-S\*') contacts). An ANOVA based on ART found a main effect of TYPE on *Manipulation Time* ( $F_{2,880} = 30.9, p < .0001$ ) but no interaction effect of SIZE  $\times$  TYPE. Post-hoc tests found manipulations with *sides* (2.29s, SD = 1.42s) faster than *edges* (3.12s, SD = 2.12s) and *corners* (3.53s, SD = 2.31s) (both  $p < .0001$ ), and *edges* faster than *corners* ( $p < .01$ ) (see bottom of Figure 5). For touch input ('T'), there was no significant effect of SIZE.

#### Comparison with Pen Manipulation Time

The pen is treated differently, as it only has five contacts and is a different shape. The edge around the nib end is used as the pen's starting contact, since this mimics the natural inking position and the pen has no corners. Pen manipulations are between the starting contact and the nib end 'NIB', cylindrical barrel 'BARREL', eraser end 'BACK', and the edge around the eraser end 'BACK EDGE'. For comparison, we consider these equivalent to Conté CONTACTS 'S-S1', 'B-S4', 'O-S1', and 'O-C1' respectively. Equivalent CONTACTS were determined using first contact location, then contact type. The pen and Conté also both include touch as a CONTACT ('T').

In most cases, Conté is comparable to the pen with the exception of Conté opposite corner (O-C1) and the edge around the pen's eraser end (BACK EDGE) (Figure 6). To find this result, we created a factor SIZE+, which combines the three-level SIZE factor with a fourth pen level. We also created a 5-level factor called CONTACT+ representing the 5 equivalent pen and Conté CONTACTS. An ART ANOVA found main effects of SIZE+ ( $F_{3,209} = 17.75, p < .0001$ ) and CONTACT+ ( $F_{4,209} = 54.01, p < .0001$ ), and an interaction between SIZE+ and CONTACT+ on *Manipulation Time* ( $F_{12,209} = 4.82, p < .0001$ ). Post-hoc tests revealed differences between the 'BACK EDGE' pen contact and the equivalent 'O-C1' Conté contact for SMALL and MEDIUM

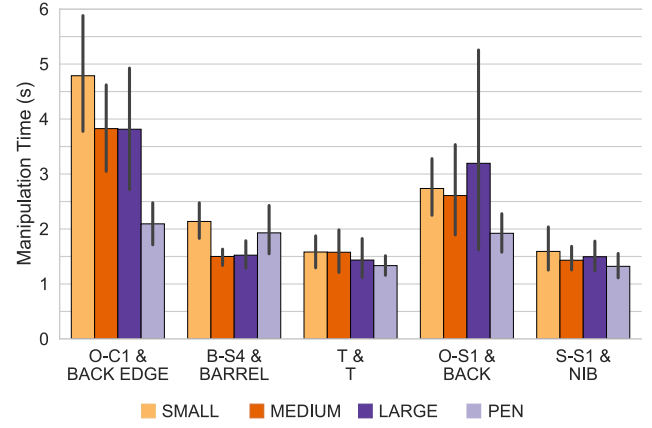


Figure 6. “pen-like” contact manipulation times for pen and Conté (error bars 95% CI).

(both  $p < .05$ ), but not LARGE. There were no other pairwise differences between PEN and equivalent Conté contacts.

## Discussion

The main finding of this experiment is that manipulation time decreases with the largest device size we tested. We hypothesized that larger devices would *increase* manipulation time due to contacts being “farther away”, and Vogel and Casiez suggested small form factors would be easier to manipulate. We found the opposite. We believe this can be explained by smaller devices being more difficult to grip. The implication is that our large-sized Conté device with internal sensing hardware should not impact manipulation performance. Given our results, a follow up study could explore additional sizes, such as sizes between medium and large, even larger sizes, and other cuboid dimensions.

Other results confirm that manipulation time is lower for nearby contacts, independent of size. We also observed that sides are faster than edges and corners. Last, we did not observe a difference between a Conté device and a pen for four of five equivalent contact manipulations. There is some manipulation time cost when using Conté compared to a pen overall, but given the increase in usable contact-to-command mappings, this seems like a worthwhile trade-off. The question remains if users can actually learn and remember so many contact-to-command mappings.

## EXPERIMENT 2: LEARNING

Manipulating a Conté device to select a command assumes the user can remember what each contact does. The goal of this experiment is to formally investigate learning and retention of the associated 3D spatially mapped commands. Two levels of difficulty are evaluated, 9 commands mapped to contacts at one end, and 26 commands mapped to all possible contacts. An on-screen guide can be activated to see the command-to-contact mapping. As a baseline comparison, a “Grid” 2D spatial interface (based on FastTap [5, 11]) is also tested. We also use the same “Shape Slicer” game and training method as the FastTap experiments.

We expected people could learn and retain 9 commands in Conté and Grid. Previous FastTap studies showed 12 commands are reasonable to learn if spatially mapped. In the Conté 9 command condition, all 9 commands are located on one end, so large end-over-end device rotations are not required, and the 3D spatial mapping is more stable. However, we expected it would be harder to remember 26 commands with Conté. Grid still has a stable 2D spatial mapping, but the 3D spatial mapping is continually re-oriented in space when a Conté device is flipped end-over-end.

### Participants

We recruited 32 participants for this between subjects experiment, 8 per condition (mean age 27.2, SD = 5.9, 25 men, 7 women, 6 left-handed). Remuneration was \$20 for two-hour 26-command conditions, and \$10 for one-hour 9-command conditions. Adding more participants may reinforce existing significant differences and reveal more subtle ones, but our goal is not to beat Grid, but to use it as a relative comparison.

### Apparatus

Our Conté device was used with a 13.3" (1920 × 1080 px, 165 PPI) touch-screen laptop (Windows 10). The laptop screen was placed flat on a table with the laptop rotated 180°, and an external keyboard used (Figure 7). Note the functional prototype is needed to enable the more complex interaction and real capacitive display.

### Procedure

The task is to invoke the command indicated by an icon falling from the top of the screen at a constant speed. In each trial, the icon appears at a random horizontal location, and falls with an experimentally controlled speed. Command invocation is accomplished by drawing or tapping with the Conté device. If the wrong command was used, the border of the icon turned red indicating an error, and the border turned green if the right command was used. As long as the icon did not reach the bottom, participants could try to complete the trial again. During the trial, they can optionally activate a guide to see the command mappings, but the guide must be closed before invoking a command; this is enforced by the software.

The 26 commands are represented as icons in a drawing application: 4 pencil sizes; 4 brush sizes; 3 eraser sizes; 8 colours (different coloured paint cans); and 7 general commands (delete, open, save, undo, redo, copy, paste). The 11 pencil, brush, and eraser commands are invoked by dragging

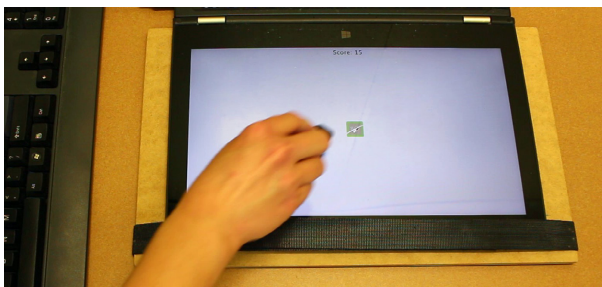


Figure 7. Touchscreen laptop, external keyboard, and game.

the device through the falling icon. The path is rendered in a style matching the command, such as black lines of different widths for the 4 pencil commands. The other 15 commands are invoked with a tap anywhere on the screen, with the corresponding command icon shown at the point of contact for 2 seconds.

A score provides behavioural incentive. Each completed trial is awarded 50 points. To deter guessing, 15 points are deducted for each error (up to a maximum of 45 points per trial) and for the icon reaching the bottom. To deter over reliance on the command guide, 10 points are deducted when it is triggered one or more times in a trial. Our scoring system is more fine-grained than the original Shape Slicer game [5].

When testing recall, the target command icon is centred on the screen with no movement or time limit. The guide is also disabled, there is no score, and no feedback is provided beyond confirming that a command was invoked (but not *which* command). There was no distinction between drawing or tapping commands, and each recall task trial ended immediately after command invocation. Videos of Conté test blocks were reviewed to verify that the contact reported by the device was correct: 71 trials (8.8%) were manually corrected. Due to a logging error, one test block (26 trials) was only classified by video. We discuss recognition accuracy again below.

### Conté Technique

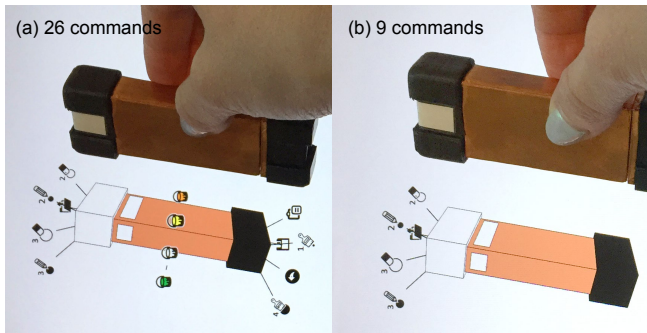
The Conté device works as described above. To enable discovery of command-to-contact mappings, there is an on-screen guide showing a mirrored view of the device as a 3D rendering, with non-occluded contacts labelled with command icons (Figure 8). Pilot studies indicated this was easier to use than a non-mirrored guide. Holding the ‘m’ on an external keyboard activates the guide after a delay of 500ms, and releasing ‘m’ dismisses it. Adding a delay is thought to promote novice-to-expert learning [9]. Outside of an experiment, the guide could be activated by a gesture like shaking or a finger tap on the device’s side. Our focus is not on guide activation techniques, so a simple key press avoids confounds. The guide is used while the device is held in the air, and the guide orientation matches the device in real time using the IMU data stream. To correct for IMU drift, yaw angle is reset by pressing ‘n’ while the device is held in a known orientation.

For the 26-command version, the 4 pen commands are mapped to corners on one end, and the 4 brush commands mapped to four corners on the other end. The 3 eraser commands and delete are mapped to edges on one end, and copy, paste, undo, redo are mapped to the four edges of the opposite end. Open and save are mapped to the end sides, and the 8 colours are mapped to barrel edges and sides. The 9-command version used only commands on one end.

### Grid Technique

Although FastTap was designed for touch, our Grid technique uses the Conté device for selecting and invoking commands to avoid an input method confound. The device contact recognition system was running during the Grid condition to avoid differences in responsiveness. Our goal is to examine learning and recall, we are not comparing Conté input with touch input.

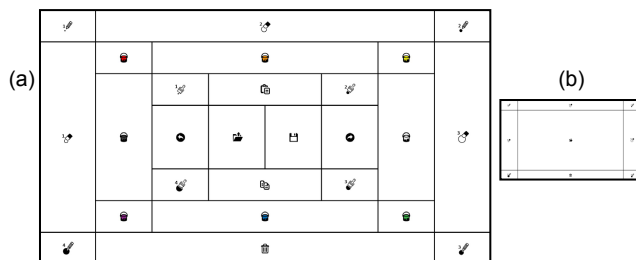




**Figure 8. Conté on-screen guide showing command-to-contact mapping: (a) 26-commands; (b) 9-commands. White marks create 3D landmarks.**

Like FastTap, the Grid technique has three steps. First, a selection mode is activated by pressing and holding the ‘m’ key. Second, any corner or end edge of the Conté device is used to tap a position corresponding to the desired command’s grid cell. Third, the selected command is invoked by either dragging through the falling icon, or tapping again anywhere on the screen, depending on command type. A guide showing the grid of commands is displayed if no selection occurs within 500ms (Figure 9). Feedback is identical to the Conté technique, with the addition of highlighting the selected grid cell in blue until the ‘m’ key is released.

We designed the Grid interface to have a strong semantic grouping and layout that is directly comparable to Conté. For the 26-command version, four nested rings of rectangles are used (Figure 9a). The outer ring corresponds to Conté corners and edges of one end; the second ring corresponds to Conté barrel sides and edges; the third ring corresponds to Conté corners and edges at the other end. The innermost cell is split with each half corresponding to a Conté end side. The order of commands in each ring is the same as on Conté. The 9-command version is the same size, but only the outer ring and single inner cell are used (Figure 9b).



**Figure 9. Grid guide with: (a) 26-commands; (b) 9-commands.**

## Design

The primary independent variable is **CONDITION** with four levels representing each combination of technique and command size: CONTE-26, CONTE-9, GRID-26, and GRID-9. Each participant completed 24 blocks of training, where each block presented one trial per command, in random order. Following Lafreniere et al.’s method for effective skill transfer [11], blocks are grouped into 6 *speed stages* with 4 blocks each. The time for an icon to fall is determined by the 6 stages, in this

order: slow (20s), medium (10s), fast (5s), medium (10s), fast (5s), medium (10s). Before training began, each participant was briefed on the game and score, and shown how to use the technique with, and without the guide. They could take a break between stages.

After all training blocks, participants immediately completed a test block of recall trials. Then they completed a post-experiment survey and interview for 10-minutes, followed by another recall test block. Finally, they returned for a third recall test block after approximately 24 hours. Before training began, participants were told to expect recall tests, but they were not given any time to look at the guide or refresh their memory before test blocks.

This is a mixed design with **CONDITION** a between-subjects factor and training **BLOCK** a repeated measure. The experiment took 1 to 2 hours, depending on condition.

## Results

Retention is examined directly using recall rates for the three test blocks, and learning relates to how performance changes during training. We examine completion time, expert usage, and error rate during all training blocks, paying special attention to the final stage. This stage has a less urgent medium speed, so it captures more realistic usage [11]. 316 (2.4%) trials more than 3 standard deviations from each **CONDITION** mean were removed as outliers.

Unless stated otherwise, in the analysis below a repeated measures ANOVA on log-transformed data is used when residuals are log-normally distributed. When residuals do not conform to a known distribution, an ANOVA based on ART [28] is used if multiple factors are present, and a Kruskal Wallis test is used on analyses of one between-subjects factor. Both are followed by post hoc Mann-Whitney U tests corrected with Holm’s sequential Bonferroni procedure.

## Recall

This is the proportion of correct command selections in a test block. We treat recall in each block as a distinct dependent variable: *Recall 0-min*, *Recall 10-min*, and *Recall 24-hr*. Overall, Conté recall was comparable or higher than Grid. Even after 24 hours, participants could recall slightly more than 24 out of 26 Conté commands on average (Figure 10).

For tests right after training and after 10 minutes, recall rates for Conté and Grid do not differ significantly regardless of command size (Figure 10). A Kruskal-Wallis test did not find an effect of **CONDITION** on *Recall 0-min* or *Recall 10-min*. For *Recall 0-min*, measured recall rates are: 93% for CONTE-26; 99% for CONTE-9; 92% for GRID-26; and 96% for GRID-9. For *Recall 10-min* they are: 94% for CONTE-26; 100% for CONTE-9; 92% for GRID-26; and 99% for GRID-9.

For recall tests after 24 hours, there is a significant difference between the two Grid conditions, but there is not a significant difference between any other pair of conditions. A main effect of **CONDITION** on *Recall 24-hr* was significant ( $\chi^2_{(3)} = 9.63$ ,  $p < .05$ ). Post hoc tests show GRID-9 (100%, SD = 0%) is higher than GRID-26 (88%, SD = 15.76%) ( $p < .05$ ). Measured



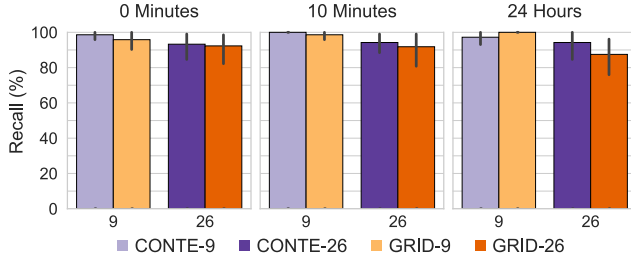


Figure 10. Recall after classification error correction (error 95% CI).

recall rates for Conté after 24 hours were 97% (SD = 5.14%) for CONTE-9 and 94% (SD = 11.45%) for CONTE-26.

### Completion Time

Completion time was measured using only successful trials, from when an icon began to fall until the correct command was invoked. This includes time to recover from user or device errors, and time to view the guide. All conditions have highest completion times in the initial slow stage and have slightly higher completion times in medium stages than fast stages. Examining blocks in the last stage only, there was no learning effect and times for conditions were comparable. An ANOVA on log-transformed data found no  $\text{CONDITION} \times \text{BLOCK}$  interaction, and no significant effect of  $\text{CONDITION}$  or  $\text{BLOCK}$  on *Completion Time*.

### Expert Usage Rate

This is defined as the proportion of trials in which the guide is not triggered. Adoption of expert techniques by the end of training indicates the participant is likely to continue to use it in a time-independent task [11]. Over all blocks, CONTE-26 and CONTE-9 exhibit an increasing trend in expert use, and expert mode is used exclusively by the final stage. Expert use of GRID-26 and GRID-9 increases significantly during the first stage, but neither are ever completely adopted (Figure 12).

Considering only the final medium stage, participants make expert selections more often with Conté. An ANOVA based on ART found main effects of  $\text{BLOCK}$  ( $F_{3,88} = 15.97, p < .0001$ ) and  $\text{CONDITION}$  ( $F_{3,28} = 5.44, p < .01$ ) on *Expert Usage Rate*, and a significant  $\text{CONDITION} \times \text{BLOCK}$  interaction ( $F_{9,84} = 4.07, p < .001$ ). Expert use with CONTE-26 was second highest (98.1%, SD = 3.9%), and CONTE-9 reached (100.0%, SD = 0%), both

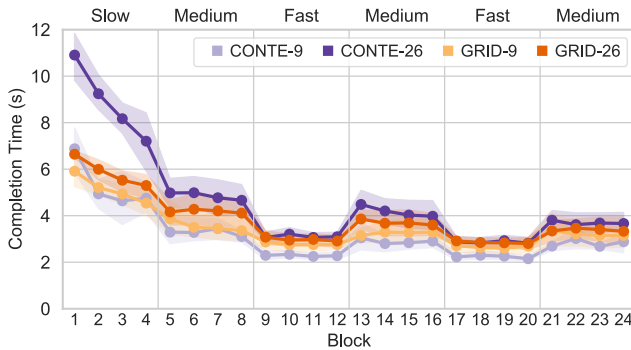


Figure 11. Completion Time by BLOCK (ribbons 95% CI).

higher than GRID-26 (82.3%, SD = 33%) ( $p < .01$ ) and GRID-9 (65.6%, SD = 45%) (all  $p < .05$ ). The main effect of  $\text{BLOCK}$  does not indicate a learning effect since expert rate did not increase over blocks, and the interaction with  $\text{BLOCK}$  can be explained by the larger variation in expert rate between blocks in both Grid conditions, particularly GRID-9.

### Error Rate

This is defined as the proportion of trials with one or more errors, including user errors and device errors. Grid conditions rely on a professionally manufactured laptop which we assume causes a negligible number of errors, while Conté conditions rely on our novel, lab-made device and contain a larger proportion of device errors.

Over all blocks, CONTE-9, GRID-9, and GRID-26 have error rates below 22%. CONTE-26 has higher error rates throughout, but stabilizes near 20% in the final medium stage. In the final medium stage, there was no learning effect, and error rates are higher with Conté. We found higher Conté error rates than in *Recall 24-hr* (CONTE-26: 19.8% vs. 6%; CONTE-9 15.3% vs. 3%). Retention tests were corrected using video footage and represent *true* user error rates, but the game itself was too fast-paced and too long to do the same. An ART ANOVA showed there was a main effect of  $\text{CONDITION}$  ( $F_{3,28} = 20.73, p < .0001$ ) on *Error Rate*, but no main effect of  $\text{BLOCK}$  and no interaction. GRID-9 has the lowest error rate (2.4%, SD = 5.5%), significantly lower than CONTE-26 (19.8%, SD = 11.9%) and CONTE-9 (15.3%, SD = 10.8%) (both  $p < .0001$ ). GRID-26 has the second lowest error rate (5.2%, SD = 5.8%), significantly lower than CONTE-26 ( $p < .0001$ ) and CONTE-9 ( $p < .01$ ).

### Discussion

To determine the impact of having a small number of participants per group, we performed a power analysis which reveals the magnitude of difference we could have detected with a Mann-Whitney U test. An effect size ( $r = 0.38$ ) is computed from the test statistic ( $Z = 1.52, p = 0.1363$ ) [29], and an independent samples t test power analysis, used to estimate the observed power of the non-parametric Mann-Whitney U test, finds adequate power to have detected a medium to large significant difference ( $\text{power} = 0.24, r = 0.67, n = 8, p = .05$ ) [29]. While the small number of participants is a limitation, any undetected differences between conditions are likely small. A power analysis cannot account for a confound in participants'

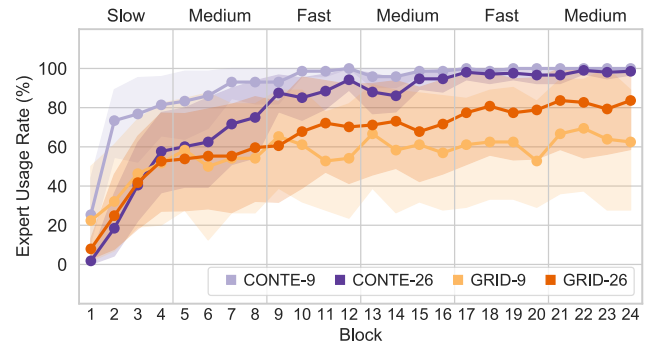


Figure 12. Expert Usage Rate by BLOCK (ribbons 95% CI).

cognitive abilities, but conditions were assigned randomly, so this seems unlikely.

The 26-command Conté has higher retention rates than expected, with an average of 94% after 24 hours, which is slightly higher than the grid condition. This suggests that tangible pen-like input devices are a worthwhile alternative selection technique to 2D interfaces like FastTap, with additional benefits of integrating command selection with parameter manipulation.

Completion time in the final stage was similar for all conditions, which includes recovery time from user errors and time to use the guide. Conté completion time also includes recovery from device errors. It is possible that without device errors Conté conditions would have been faster than Grid.

Grid expert usage was lower than Conté in the final stage, which is in contrast to Lafreniere et al.'s FastTap results [11]. Our grid technique is different since it is bimanual, has more commands, and uses a larger screen. We mitigate this with a longer novice delay, but the temporal cost of Conté novice selection (4.7s for 26 commands, 4.0s for 9) is still higher than grid novice selection (1.7s for 26 commands, 1.2s for 9), and might provide an incentive to use Conté expert mode. Using a similar training game, Lafreniere et al. found expert usage persists when the task is changed and speed requirement is removed if consistent expert selection is established in training [11]. Since Conté expert usage rates are similar to Lafreniere et al.'s, similar post-training persistence is expected.

A high device error rate may have prevented us from measuring the true, possibly higher, recall rate. We reviewed video logs to manually correct device misclassification errors in 8.8% of test trials. A similar device error rate likely occurred during training and contributed to high Conté error rates, and may have made learning more challenging. Using the 8.8% device error rate in test blocks, we estimate user error rates in the final training stage: Conté-26 11% (originally 19.8%), Conté-9 6.5% (originally 15.3 %). Conté conditions still have higher estimated user error rates than Grid, Grid-26 5.2% and Grid-9 2.4%, but the difference is much smaller. Time constraints during training were not present during test blocks and may account for remaining differences.

Of the 71 test trials corrected for device errors, 68 were “near misses” to adjacent contacts. All adjacent contacts have different geometry, and future work could combine size and shape data from a touch sensor with IMU data to increase classification accuracy. This would improve feedback, could make learning easier, and may result in even higher recall.

## DESIGN IMPLICATIONS

It is important to recognize that the Conté device belongs to a broader category of input devices able to detect which of their geometric features contacts a surface. The 3D cuboid shape we test is one of the simplest instances, and arguably the most elemental and generalizable, but there are many possible variations. For example, the shape we tested has a minimal amount of visual and geometric asymmetry — the contacts have the smallest possible amount of distinction, just enough to create 3D landmarks. This means that our results are likely a lower bound for learning and recall performance.

Visual, geometric, and passive haptic landmarks can help define a frame-of-reference when manipulating tangible pen-like devices in 3D. Designing a device using more varied visual markings on each surface, and making the shape itself more asymmetrical, such as extruding a trapezoid, perhaps with one end a pointed cone or rounded hemisphere, may make learning and recall even easier. Adding different passive haptic textures on different sides, such as patterns of bumps or ridges, would also increase available landmarks. Other related ideas include controlling the centre of mass to create a heavy end or side, and making some edges more pliable than others [23]. Diverse tactile patterns and asymmetries may also enable more reliable eyes-free usage and reduce error rate.

Our manipulation experiment showed a Conté device needs to be large enough to facilitate grip. Commands with similar semantic meaning should be grouped together on nearby contacts. Results from the manipulation experiment further indicate that frequently used commands should be positioned in priority on the same side, while less frequently used commands can be positioned on the barrel or opposite side. Further refining contact-to-command associations depends on the degree of control needed by each command (*e.g.* a corner for writing) and possible additional landmarks if some contacts are permanently mapped to commands, such as a rubber feeling on a side mapped to an eraser command.

## CONCLUSION

The original Conté paper was about the *idea* of a pen-like tangible input device. It demonstrated how transitioning to different contacts enabled a range of interactions like a multipurpose pen, an interactive ruler, a temporary mouse, and more. Now, our results show this type of device is practical in two key aspects that were never tested. It has contact-to-contact manipulation times comparable to a pen, and people can learn and recall a large portion of the 26 possible contact-to-command mappings using its 3D spatial layout. The latter is comparable to an equivalent grid interface using a 2D spatial layout. On a technical level, we designed and implemented a new version of Conté which realizes Vogel and Casiez's vision of a self-contained hardware device that detects all 26 contacts and works with standard touch displays. This re-imagined device is a significant improvement on their 10-contact prototype that required a specialized tabletop.

Yet, our work contributes more than a validation of the original Conté idea. Our investigation of size-dependent manipulation, and our comparison of learning with 2D and 3D spatial mappings, are higher level questions we believe have not been tested in this context. These results open up directions for future work, for example examining how these findings generalize to other related devices and styles of tangible input.

## ACKNOWLEDGEMENTS

This work made possible by NSERC Discovery Grant 2018-05187, the Canada Foundation for Innovation Infrastructure Fund 33151 “Facility for Fully Interactive Physio-digital Spaces,” and Ontario Early Researcher Award ER16-12-184. Thanks to Jacob O. Wobbrock and Yim Register.

## REFERENCES

1. Fraser Anderson and Walter F. Bischof. 2013. Learning and Performance with Gesture Guides. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1109–1118. DOI: <http://dx.doi.org/10.1145/2470654.2466143>
2. George W. Fitzmaurice, Hiroshi Ishii, and William A. S. Buxton. 1995. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 442–449. DOI: <http://dx.doi.org/10.1145/223904.223964>
3. Brigitte N Frederick. 1999. Fixed-, Random-, and Mixed-Effects ANOVA Models: A User-Friendly Guide for Increasing the Generalizability of ANOVA Results. (1999), 1–23.
4. Karen Grace-Martin. 2008-2019. Six Differences Between Repeated Measures ANOVA and Linear Mixed Models. <https://www.theanalysisfactor.com/>. (2008-2019).
5. Carl Gutwin, Andy Cockburn, and Benjamin Lafreniere. 2015. Testing the Rehearsal Hypothesis with Two FastTap Interfaces. In *Proceedings of the 41st Graphics Interface Conference (GI '15)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 223–231. <http://dl.acm.org/citation.cfm?id=2788890.2788930>
6. Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. 2014. Faster Command Selection on Tablets with FastTap. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2617–2626. DOI: <http://dx.doi.org/10.1145/2556288.2557136>
7. FiftyThree Inc. 2017. Pencil Stylus for iPad, iPad Pro & iPhone. <https://www.fiftythree.com/pencil>. (2 2017).
8. Hiroshi Ishii and Brygg Ullmer. 1997. Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '97)*. ACM, New York, NY, USA, 234–241. DOI: <http://dx.doi.org/10.1145/258549.258715>
9. Gordon Kurtenbach. 1993. *The design and evaluation of marking menus*. University of Toronto.
10. Gordon Kurtenbach and William Buxton. 1994. User Learning and Performance with Marking Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*. ACM, New York, NY, USA, 258–264. DOI: <http://dx.doi.org/10.1145/191666.191759>
11. Benjamin Lafreniere, Carl Gutwin, and Andy Cockburn. 2017. Investigating the Post-Training Persistence of Expert Interaction Techniques. *ACM Trans. Comput.-Hum. Interact.* 24, 4, Article 29 (Aug. 2017), 46 pages. DOI: <http://dx.doi.org/10.1145/3119928>
12. Yang Li, Ken Hinckley, Zhiwei Guan, and James A. Landay. 2005. Experimental Analysis of Mode Switching Techniques in Pen-Based User Interfaces. (2005), 461–470. DOI: <http://dx.doi.org/10.1145/1054972.1055036>
13. Microsoft. 2017. Microsoft Surface Pen. <https://www.microsoft.com/en-us/surface/accessories/surface-pen?icid=MSCOMSearchTopResultSurfacePen>. (9 2017).
14. Halla B. Olafsdottir, Theophanis Tsandilas, and Caroline Appert. 2014. Prospective Motor Control on Tablets: Planning Grasp for Multitouch Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2893–2902. DOI: <http://dx.doi.org/10.1145/2556288.2557029>
15. Gary Perelman, Marcos Serrano, Christophe Bortolaso, Celia Picard, Mustapha Derras, and Emmanuel Dubois. 2019. Combining Tablets with Smartphones for Data Analytics. In *IFIP Conference on Human-Computer Interaction*. Springer, Springer International Publishing, 439–460. DOI: [http://dx.doi.org/10.1007/978-3-030-29390-1\\_24](http://dx.doi.org/10.1007/978-3-030-29390-1_24)
16. Gary Perelman, Marcos Serrano, Mathieu Raynal, Celia Picard, Mustapha Derras, and Emmanuel Dubois. 2015. The Roly-Poly Mouse: Designing a Rolling Input Device Unifying 2D and 3D Interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 327–336. DOI: <http://dx.doi.org/10.1145/2702123.2702244>
17. Simon T. Perrault, Eric Lecolinet, Yoann Pascal Bourse, Shengdong Zhao, and Yves Guiard. 2015. Physical Loci: Leveraging Spatial, Object and Semantic Memory for Command Selection. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 299–308. DOI: <http://dx.doi.org/10.1145/2702123.2702126>
18. Jun Rekimoto and Eduardo Sciammarella. 2000. ToolStone: Effective Use of the Physical Manipulation Vocabularies of Input Devices. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST '00)*. ACM, New York, NY, USA, 109–117. DOI: <http://dx.doi.org/10.1145/354401.354421>
19. Joey Scarr, Andy Cockburn, and Carl Gutwin. 2013a. Supporting and Exploiting Spatial Memory in User Interfaces. *Found. Trends Hum.-Comput. Interact.* 6, 1 (Dec. 2013), 1–84. DOI: <http://dx.doi.org/10.1561/11000000046>

20. Joey Scarr, Andy Cockburn, Carl Gutwin, and Sylvain Malacria. 2013b. Testing the Robustness and Performance of Spatially Consistent Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 3139–3148. DOI : <http://dx.doi.org/10.1145/2470654.2466430>
21. Hyunyoung Song, Hrvoje Benko, Francois Guimbretiere, Shahram Izadi, Xiang Cao, and Ken Hinckley. 2011. Grips and Gestures on a Multi-touch Pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1323–1332. DOI : <http://dx.doi.org/10.1145/1978942.1979138>
22. Minghui Sun, Xiang Cao, Hyunyoung Song, Shahram Izadi, Hrvoje Benko, Francois Guimbretiere, Xiangshi Ren, and Ken Hinckley. 2011. Enhancing Naturalness of Pen-and-tablet Drawing Through Context Sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11)*. ACM, New York, NY, USA, 83–86. DOI : <http://dx.doi.org/10.1145/2076354.2076371>
23. Cesar Torres, Tim Campbell, Neil Kumar, and Eric Paulos. 2015. HapticPrint: Designing Feel Aesthetics for Digital Fabrication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 583–591. DOI : <http://dx.doi.org/10.1145/2807442.2807492>
24. Md. Sami Uddin, Carl Gutwin, and Andy Cockburn. 2017. The Effects of Artificial Landmarks on Learning and Performance in Spatial-Memory Interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3843–3855. DOI : <http://dx.doi.org/10.1145/3025453.3025497>
25. Md. Sami Uddin, Carl Gutwin, and Benjamin Lafreniere. 2016. HandMark Menus: Rapid Command Selection and Large Command Sets on Multi-Touch Displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5836–5848. DOI : <http://dx.doi.org/10.1145/2858036.2858211>
26. Kristof Van Laerhoven, Nicolas Villar, Albrecht Schmidt, Gerd Kortuem, and Hans Gellersen. 2003. Using an Autonomous Cube for Basic Navigation and Input. In *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI '03)*. ACM, New York, NY, USA, 203–210. DOI : <http://dx.doi.org/10.1145/958432.958472>
27. Daniel Vogel and Géry Casiez. 2011. Conté: Multimodal Input Inspired by an Artist's Crayon. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 357–366. DOI : <http://dx.doi.org/10.1145/2047196.2047242>
28. Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 143–146. DOI : <http://dx.doi.org/10.1145/1978942.1978963>
29. Koji Yatani. 2016. Effect sizes and power analysis in HCI. In *Modern Statistical Methods for HCI*. Springer, 87–110.